École polytechnique — X2020 — INF412 Fondements de l'informatique

PC6 notée

(corrigé)

Cet énoncé comporte trois parties indépendantes et qui pourront être résolues dans n'importe quel ordre. Dans chaque partie, on pourra, pour répondre à une question, admettre les résultats dont on demande la démonstration aux questions *précédentes*. Il n'est pas nécessaire de traiter toutes les questions pour avoir la note maximale. Les correcteurs vous remercient d'avance d'écrire lisiblement.

1 Ensembles partiellement ordonnés

Dans cette partie, quand on définira une théorie, on prendra soin de bien en spécifier la signature, et de vérifier que les axiomes sont sur cette signature.

On rappelle qu'un *ensemble partiellement ordonné* $(X, \leq)$ est un ensemble X muni d'une relation $\leq$ qui est réflexive, transitive et antisymétrique.

Question 1.1. *Donner une théorie $\mathcal{T}$ dont les modèles égalitaires sont les ensembles partiellement ordonnés.*

Solution : Sur la signature avec les symboles de relation binaires $\leq$ et $=$, on considère la théorie avec les axiomes suivants.

— La relation est réflexive :

$$\forall x.x \leq x$$

— La relation est transitive :

$$\forall x.\forall y.\forall z.x \leq y \wedge y \leq z \Rightarrow x \leq z$$

— La relation est antisymétrique :

$$\forall x.\forall y.x \leq y \wedge y \leq x \Rightarrow x = y$$

□

Question 1.2. *Donner une théorie dont les modèles sont les ensembles partiellement ordonnés avec un élément maximal.*

Solution : On ajoute à la théorie $\mathcal{T}$ l'axiome

$$\exists x.\forall y.y \leq x$$

□

Question 1.3. *Est-ce que les formules suivantes sont prouvables dans la théorie $\mathcal{T}$? Justifiez.*

1. $\forall x.\forall y.\forall z.[(x \leq y) \wedge (y \leq z) \wedge (z \leq x)] \Rightarrow x = y$.
2. $\forall x.\exists y.(y \leq x) \wedge \neg(y = x)$.

Solution :

1. Oui. On peut formaliser la preuve suivante. Supposons donnés x, y, z tels que $x \leq y$, $y \leq z$ et $z \leq x$. Par transitivité, on a $y \leq x$ et par antisymétrie on en déduit $x = y$.
2. Non, par validité. Si cette formule était prouvable, elle serait vraie dans tout modèle. Or, l'ensemble partiellement ordonné $\mathbb{N}$ ne vérifie pas cette formule car pour 0 il n'existe pas d'élément strictement plus petit.

□

Question 1.4. *La théorie $\mathcal{T}'$ est obtenue à partir de $\mathcal{T}$ en ajoutant l'axiome*

$$\forall x.\forall z.\neg(x = z) \Rightarrow [\exists y.(x \leq y) \wedge \neg(x = y) \wedge (y \leq z) \wedge \neg(y = z)]$$

Est-elle cohérente ? Justifiez.

Solution : Oui. Un modèle de $\mathcal{T}'$ est un ensemble partiellement ordonné qui satisfait

$$\forall x.\forall z.x \neq z \Rightarrow (\exists y.x < y \wedge y < z)$$

(c'est la formule ci-dessus réécrite avec des notations plus habituelles). Un tel modèle est par exemple $\mathbb{Q}$ (ou $\mathbb{R}$) avec l'ordre usuel. En effet, pour $x \neq z$, on peut prendre $y = (x + z)/2$ et on a $x < y < z$. Par le théorème de complétude, la théorie $\mathcal{T}'$ est donc cohérente. □

Étant donné un ensemble partiellement ordonné $(X, \leq)$, une *chaîne* est une suite $x_0, x_1, x_2, \ldots$ finie ou infinie d'éléments de X qui est strictement décroissante

$$x_0 > x_1 > x_2 > \ldots$$

Question 1.5. *Étant donné un entier $n \in \mathbb{N}$, donner une théorie dont les modèles égalitaires sont les ensembles partiellement ordonnés possédant une chaîne de longueur au moins n.*

Solution : On part de la théorie $\mathcal{T}$ et on ajoute l'axiome C_n

$$\exists x_1.\exists x_2 \ldots \exists x_n.x_1 > x_2 \wedge x_2 > x_3 \wedge \ldots \wedge x_3 > x_n$$

qui impose qu'il existe une chaîne de longueur au moins n. Ici, on utilise le symbole $>$ qui n'est pas dans la signature, mais on peut remplacer $x > y$ par $(y \leq x) \wedge \neg(x = y)$. □

Un ensemble partiellement ordonné est *bien fondé* lorsqu'il n'admet pas de suite infinie strictement décroissante.

Question 1.6. *Donner une théorie dont les modèle égalitaires sont les ensembles partiellement ordonnés possédant une chaîne de longueur au moins n, pour tout $n \in \mathbb{N}$. Montrer que cette théorie admet un modèle bien fondé.*

Solution : On prend la même théorie qu'à la question précédente, en prenant tous les C_n à la fois. Un modèle de cette théorie est l'ensemble $\mathbb{N}$ avec l'ordre habituel. Cet ensemble est bien fondé et, pour tout $n \in \mathbb{N}$, on a une chaîne de longueur n :

$$n - 1 > n - 2 > n - 3 > \ldots > 2 > 1 > 0$$

□

Question 1.7. *Montrer qu'il n'existe pas de théorie du premier ordre dont les modèles égalitaires sont les ensembles partiellement ordonnés bien fondés (on pourra supposer que la signature contient des symboles de relations binaires $\leq$ et $=$). Merci de soigner la rédaction de cette question.*

Solution : Supposons qu'il existe une telle théorie $\mathcal{T}$. On considère la théorie $\mathcal{T}'$ obtenue en ajoutant un nombre dénombrable de constantes c_i (avec $i \in \mathbb{N}$) et les axiomes A_i

$$c_i > c_{i+1}$$

pour tout entier i. Montrons par le théorème de compacité que $\mathcal{T}'$ admet un modèle. Considérons un sous-ensemble fini $\mathcal{T}'' \subseteq \mathcal{T}'$ et notons i_0 le plus grand indice tel que $A_{i_0} \in \mathcal{T}''$. Tout ensemble partiellement ordonné avec une chaîne de longueur $i_0 + 1$ est un modèle, et donc $(\mathbb{N}, \leq)$ est un modèle par la question précédente. Plus précisément, on peut interpréter les constantes par

$$[\![c_j]\!] = \begin{cases} i_0 - j & \text{si } j \leq i_0, \\ 0 & \text{sinon.} \end{cases}$$

Par le théorème de compacité, on a donc un modèle de $\mathcal{T}'$ et ce modèle n'est pas bien fondé car l'interprétation des c_i nous fournit une chaîne infinie. Comme $\mathcal{T} \subseteq \mathcal{T}'$, ce modèle est aussi un modèle de $\mathcal{T}$, en contradiction avec l'hypothèse que $\mathcal{T}$ n'admet que des ensembles partiellement ordonnés bien fondés comme modèles. $\square$

2 Problèmes de décision et machines de Turing

Question 2.1. *Montrer que le problème suivant est décidable : « déterminer si une machine de Turing M est telle que $L(M)$ est reconnu par une (autre) machine de Turing qui accepte en ayant fait autant de déplacements vers la gauche que de déplacements vers la droite ».*

Solution : Pour toute machine M, on peut construire une machine M' qui reconnaît le même langage et a cette propriété, ce qui rend ce problème décidable (car la réponse est toujours oui). En effet, il suffit d'ajouter des symboles à l'alphabet pour se souvenir de où est la tête de lecture et où est la case sur laquelle le calcul a commencé. A chaque étape, la machine M' scanne le ruban pour trouver où la tête de M devrait être, fait la transition comme dans M, et retourne à la première case. Lorsqu'elle retourne à la première case, elle a effectué autant de déplacements vers la gauche et vers la droite, elle peut donc accepter à ce moment là pour satisfaire la condition de l'énoncé. $\square$

Question 2.2. *Le problème suivant est-il décidable ? semi-décidable ? : « déterminer si une machine de Turing M accepte au moins un mot dont la longueur est un nombre premier ».*

Solution : C'est indécidable par le théorème de Rice : la propriété d'accepter un mot dont la longueur est un nombre premier est une propriété du langage, qui est satisfaite par Σ^* mais pas par $\varnothing$. De plus, ce problème est semi-décidable car étant donné M, le langage $L(M)$ des mots qu'elle reconnaît est semi-décidable donc récursivement énumérable. Il suffit donc de lister un à un des mots acceptés jusqu'à en trouver un dont la longueur est un nombre premier. $\square$

Dans la suite, on pourra se servir du fait, démontré dans le cours, que le problème suivant n'est pas semi-décidable : « déterminer si une machine M et un mot w sont tels que M ne s'arrête pas sur w ».

Question 2.3. *Montrer que le problème suivant est indécidable et non semi-décidable : « déterminer si une machine de Turing M accepte toutes les entrées ».*

Solution : Il s'agit clairement d'une propriété de langage non triviale donc c'est indécidable par Rice. On va montrer que ce n'est pas semi-décidable en réduisant depuis le problème « étant donné une machine M et un mot w, décider si M ne s'arrête pas sur w » (ce problème n'est pas semi-décidable d'après le cours). Si on a M et w, on construit la machine M' qui sur l'entrée u va simuler M sur l'entrée w pendant $|u|$ étapes et accepte seulement si la machine n'a **pas** terminé après $|u|$ étapes. On vérifie que si M ne s'arrête pas sur w alors M' accepte toutes les entrées, par contre si M s'arrête sur w alors M' va accepter au moins un entrée (en fait toutes les entrées suffisamment longues). $\square$

Question 2.4. *En déduire que le problème suivant est indécidable et non semi-décidable : « déterminer si deux machines de Turing M et M' sont telles que $L(M) \subseteq L(M')$ ».*

Solution : Le théorème de Rice ne s'applique pas car il y a deux machines au lieu d'une. Néanmoins, on voit que si ce problème était décidable, alors en prenant M la machine qui accepte toutes les entrées, on pourrait décider le problème de savoir, sur une machine de Turing M' accepte toutes les entrée. Or ce problème est indécidable par la question précédente. Le même argument montre qu'il n'est pas semi-décidable non plus. $\square$

On s'intéresse maintenant à la classe particulière des machines de Turing qui ne se déplacent jamais vers la gauche durant les calculs, c'est à dire qu'à chaque transition la machine reste sur place ou se déplace vers la droite du ruban. On va voir que cette propriété a des conséquences subtiles sur la décidabilité des problèmes.

Question 2.5. *Soit M une machine de Turing et w un mot. On suppose que M accepte w sans jamais se déplacer vers la gauche. Montrer que M accepte un mot w' tel que, pendant le calcul de la machine, la machine ne se déplace jamais vers la gauche et ne se trouve jamais deux fois dans le même état (q, σ) où q est l'état de la machine et σ le symbole sous la tête de lecture.*

Solution : Supposons que la propriété voulue n'est pas vérifiée par w : alors la machine a fait deux transitions $(q, \sigma) \mapsto (q', \sigma', d)$ avec le même couple (q, σ). Si cette situation se produit sur le mot w à la position i et à la position j, alors on peut supprimer de w le sous-mot $w[i..j[$ et la machine aura exactement le même comportement sur ce nouveau mot puisque ne pouvant aller vers la gauche, elle ne peut pas faire la différence entre ces deux entrées. Le mot $w' = w[1..i[w[j..[$ obtenu est strictement plus cours que w. En répétant ce processus un nombre fini de fois, on produit un mot w' qui a donc la propriété voulue. □

Question 2.6. *Montrer que le problème suivant est décidable : « déterminer si une machine de Turing M accepte au moins un mot sans se déplacer vers la gauche ».*

Solution : Le théorème de Rice ne s'applique pas car il s'agit d'une propriété de machine et non de langage. D'après la question précédente, si M accepte un mot sans se déplacer vers la gauche, alors elle en accepte un mot sans se déplacer vers la gauche et sans jamais répéter le couple (état,symbole). On voit donc qu'un tel mot est accepté en moins de $|Q| \cdot |\Sigma|$ étapes, ce qui est aussi une borne sur sa longueur. On peut donc énumérer tous ces mots et simuler M pendant ce temps fini pour voir s'il existe un tel mot. □

Question 2.7. *Soit L_s un langage semi-décidable et indécidable, M une machine de Turing et w un mot. On pose*

$$\mathcal{L}_{M,w} = \{u : u \in L_s \text{ et } M \text{ accepte } w \text{ en moins de } |u| \text{ étapes } \}.$$

Montrer que $\mathcal{L}_{M,w}$ est reconnu par une machine qui ne se déplace jamais vers la gauche si et seulement si M ne s'arrête pas sur l'entrée w.

Solution : En effet, si M ne s'arrête par sur w, la machine M' n'accepte jamais l'entrée car la simulation ne termine jamais en $|u|$ étapes pour n'importe quel mot u, on a donc $\mathcal{L}_{M,w} = \varnothing$ qui est trivialement reconnu par la machine qui ne bouge pas et refuse. A l'inverse, si M s'arrête en N étapes alors $\mathcal{L}_{M,w} = \{u \in L_s : |u| \geqslant N\}$. Or L_s n'est pas décidable et $\mathcal{L}_{M,w}$ est le langage L_s auquel on a enlevé un nombre fini de mots, il n'est donc toujours pas décidable et donc, en particulier, pas reconnu par une machine qui ne fait pas de déplacement vers la gauche. □

Question 2.8. *Montrer que le problème suivant est indécidable et non semi-décidable : « déterminer si une machine de Turing M est telle que $L(M)$ est reconnu par une (autre) machine de Turing qui ne fait pas de déplacements vers la gauche ».*

Solution : Il s'agit clairement d'une propriété de langage, elle est non-triviale car le langage vide a cette propriété et à l'inverse il existe des langages semi-décidables qui ne sont pas reconnus par de telles machines (n'importe quel langage semi-décidable mais indécidable fera l'affaire). Ce problème est donc indécidable par le théorème de Rice.

On va montrer que ce n'est pas semi-décidable en réduisant depuis le problème « étant donné une machine M et un mot w, décider si M ne s'arrête pas sur w » (ce problème n'est pas semi-décidable d'après le cours). On commence par se donner un langage L_s tel que dans la question précédente. Si on nous donne M et w, on construit la machine M' qui, sur l'entrée u, commence par vérifier si $u \in L_r$ et refuse si ce n'est pas le cas (ce calcul peut ne pas terminer). Si $u \in L_s$ (le calcule termine toujours dans ce cas) M' simule la machine M sur l'entrée w pendant $|u|$

étapes. Si la machine M a accepté w en moins de $|u|$ étape, alors M' accepte et sinon elle refuse. Autrement dit, on a $L(M') = \mathcal{L}_{M,w}$ avec la notation de la question précédente.

Mais alors $L(M')$ est reconnu par une machine de Turing qui ne se déplace jamais vers la gauche si et seulement si M ne s'arrête pas sur l'entrée w. Cela montre donc bien que notre problème n'est pas semi-décidable. $\square$

3 Graphes

Les deux premières questions concernent différentes notions de clôture pour une relation binaire quelconque. Rappelons que la clôture réflexive, la clôture symétrique et la clôture transitive d'une relation R sont les plus petites relations qui contiennent R est qui sont, respectivement, réflexive, symétrique ou transitive.

Question 3.1. *La clôture réflexive et la clôture symétrique d'une relation sont tous les deux exprimable dans la logique du premier ordre, dans le sens où les propositions*

1. *S est la clôture réflexive de R*
2. *S est la clôture symétrique de R*

peuvent être exprimées par des formules du premier ordre sur une signature avec trois symboles de relations binaires $=$, R, et S. Donnez une formule pour chacune.

Solution :

1. $S(x, y) \iff R(x, y) \vee x = y$
2. $S(x, y) \iff R(x, y) \vee R(y, x)$

$\square$

Question 3.2. *En revanche, la clôture transitive d'une relation n'est pas exprimable dans la logique du premier ordre, dans le sens où la proposition*

3. *S est la clôture transitive de R*

ne peux pas être exprimée par une formule du premier ordre sur une signature avec trois symboles de relations binaires $=$, R, et S. Expliquez pourquoi.

Solution : Supposons qu'il existe une formule ϕ du premier ordre dont les modèles égalitaires sont des ensembles muni d'une paire de relations R et S telles que S est la clôture transitive de R. On considère la théorie $\mathcal{T} = \{\phi\} \cup \{\psi_n \mid n \in \mathbb{N}\}$ obtenue en ajoutant deux constantes c et d et la formule

$$\psi_n = S(c, d) \wedge \neg\exists x_0 \ldots \exists x_n . c = x_0 \wedge x_n = d \wedge R(x_0, x_1) \wedge \cdots \wedge R(x_{n-1}, x_n)$$

pour chaque $n \in \mathbb{N}$. On peut vérifier que l'ensemble $\mathbb{N}$ muni des relations

$$R = \{(x, x+1) \mid x \in \mathbb{N}\}$$
$$S = \{(x, y) \mid x < y\}$$

est un modèle de chaque sous-ensemble fini $\mathcal{T}'$ de $\mathcal{T}$: S est la clôture transitive de R, et les formules ψ_n sont satisfaites car on peut toujours trouver c et d tels que $d - c > k$, où k est le plus grand indice tel que $\psi_k \in \mathcal{T}'$ (par exemple, $c = 0$ et $d = k + 1$). La théorie $\mathcal{T}$ admet donc un modèle par le théorème de compacité, ce qui contredit l'hypothèse que S est la clôture de R, puisque (c, d) est dans la clôture transitive de R si et seulement s'il existe une chaîne des éléments $x_0, \ldots, x_n$ avec $c = x_0$ et $x_n = d$ telle que $R(x_0, x_1) \wedge \cdots \wedge R(x_{n-1}, x_n)$. $\square$

Pour les questions suivantes, on définit un *graphe* comme une paire $G = (V, E)$ d'un ensemble V muni d'une relation $E \subseteq V \times V$. On supposera que V est dénombrable, ce qui revient à supposer $V \subseteq \mathbb{N}$.

On s'intéresse à la notion d'homomorphisme entre graphes : un *homomorphisme* $G \to H$ d'un graphe $G = (V_G, E_G)$ vers un autre graphe $H = (V_H, E_H)$ est une fonction $f : V_G \to V_H$ telle que pour chaque paire d'éléments x et y de V, $(x, y) \in E_G$ implique $(f(x), f(y)) \in E_H$.

On suppose d'abord que l'ensemble V est fini (c'est-à-dire que le graphe G est fini). Se donner G revient à se donner V et E.

Question 3.3. *Montrer que le problème suivant est décidable : « étant donnés deux graphes finis G et H, déterminer s'il existe un homomorphisme $G \to H$ ».*

Solution : Il suffit d'énumérer toutes les fonctions $f : V_G \to V_H$ et de vérifier s'il en existe une telle que $(x, y) \in E_G$ implique $(f(x), f(y)) \in E_H$. Puisque V_G et V_H sont finis, il y a un nombre fini de fonctions $f : V_G \to V_H$ et l'algorithme se terminera avec une réponse positive ou négative. □

On suppose maintenant que l'ensemble V est dénombrable mais pas forcément fini. Dans la suite, on suppose que tous les graphes sont décidables : un graphe G est dit *décidable* dans le cas où la relation $(x, y) \in E$ est décidable. Se donner un graphe dénombrable G revient par conséquent à se donner une machine de Turing M_G qui prend en entrée x et y, et qui accepte (respectivement refuse) si $(x, y) \in E$ (respectivement : $(x, y) \notin E$).

Question 3.4. *Montrer que le problème suivant est semi-décidable mais indécidable : « étant donnés un graphe fini G, et un graphe dénombrable H, déterminer s'il existe un homomorphisme $G \to H$ ».*

Solution : L'algorithme de semi-décision est similaire à l'algorithme de décision pour la question précédente : puisque V_G est fini et V_H est dénombrable, l'ensemble des fonctions $f : V_G \to V_H$ est dénombrable, et alors il suffit de les énumérer jusqu'à ce qu'on trouve une f telle que $(x, y) \in E_G$ implique $(f(x), f(y)) \in E_H$ (cette recherche peut ne jamais se terminer). Pour montrer que le problème est indécidable, nous construisons une réduction du problème de l'arrêt (Halting Problem). Prenons d'abord G le graphe avec deux sommets x et y et un arête (x, y). Ensuite, étant donné une machine de Turing M, on définit le graphe H comme suit :

- il y a une paire de sommets x_n et y_n pour chaque entier $n \in \mathbb{N}$;
- il y un arête (x_n, y_n) si et seulement si la machine M se termine dans n étapes.

Remarquer que le graphe H est décidable avec un ensemble dénombrable de sommets. Mais par construction, il existe un homomorphisme $G \to H$ si et seulement si la machine M se termine dans un nombre fini d'étapes $n \in \mathbb{N}$, ce qui déciderait le problème de l'arrêt. □

Question 3.5. *Montrer que le problème suivant est non semi-décidable : « étant donnés un graphe dénombrable G et un graphe dénombrable H, déterminer s'il existe un homomorphisme $G \to H$ ».*

Solution : Nous construisons une réduction du complément du Halting Problem. Étant donné une machine de Turing M, on définit G et H comme suit :

- G a un sommet n pour chaque entier $n \in \mathbb{N}$, et un arête $(n, n + 1)$ pour chaque $n \in \mathbb{N}$;
- H est *le graphe de configurations de M*, c'est-à-dire le graphe avec un sommet C pour chaque configuration $C = u\mathbf{q}v$ de M, et un arête (C, C') ssi $C \vdash C'$.

On voit que les deux graphes G et H sont dénombrables et décidables. Par construction, il existe un homomorphisme $G \to H$ si et seulement s'il y a une suite infinie de configurations $C_0 \vdash C_1 \vdash C_2 \vdash \ldots$, ce qui indiquerait que la machine M ne s'arrête pas. □